

# A Framework for the Analysis of Coordination in Global Software Development

Gamel O. Wiredu  
Interaction Design Centre  
Dept. of Computer Science and Information Systems  
University of Limerick  
Limerick, Ireland  
+353 61 202 798  
gamel.wiredu@ul.ie

## ABSTRACT

This paper attempts a conceptualization of coordination in Global Software Development (GSD) by arguing that distribution is a significant conditioner of software development that engenders distance-related, socio-cultural and technological conditioners. It is proposed that the core organising dimensions on which coordination analysis in GSD should focus are people, processes, information, technology and the interactions between them. It is also argued that these dimensions are characterized by process interdependencies, interpersonal and interunit conflicts, information uncertainties and equivocalities, technology representations, and their interrelations. The final argument is that the management of the dimensions' characteristics – which defines coordination – will be conditioned by distribution, and that the awareness of this conditioning must be central in coordination analysis. The resultant is an analytical framework that will hopefully proffer a theoretical foundation for research on coordination in GSD.

## Categories and Subject Descriptors

K.6.1 [Project and People Management]: *Management Techniques, Systems Development.*

## General Terms

Theory, Management, Human Factors.

## Keywords

Global Software Development, Coordination, Interdependencies, Conflicts, Uncertainties, Equivocalities, Technology Representations.

## 1. INTRODUCTION

Global Software Development (GSD) is a contemporary form of software development undertaken in globally distributed locations and facilitated by advanced information and communication technology (ICT), with the predominant aim of rationalizing the development process. GSD is increasingly becoming the norm in software development. It is now a dominant subject matter in both software development and globalization domains, and is

approached from perspectives such as “outsourcing” [28], “offshoring” [5], and “nearshoring” [1]. The common denominator of all these perspectives is distance, a determining parameter that significantly differentiates the challenges of distributed activities from their localised equivalents. Distribution is a significant conditioner of software development organising; and its implications on developers' actions are as critical as the coordination of those actions and the analytical techniques for studying that coordination.

The global distribution of software development is increasing its complexity. Kraut and Streeter [14] talked about the implications of increasing complexity of software development when they wrote about coordination and argued as follows:

“While there is no single cause of the software crisis, a major contributor is the problem of coordinating activities while developing large software systems. We will argue that this coordination becomes much more difficult as project size and complexity increase” [p.69].

In another paper [40] where I discussed the connotations of this difficulty, I argued that coordination is the key challenge of GSD and must occupy the forefront of research on the organisation of GSD. Although the global distribution of software development poses several other problems, the greatest organising challenge it presents is coordination of the interactions between distributed people, processes, information and technology. And, therefore, there is the need for greater insights into this problem that will serve as guidelines for action for both researchers and practitioners.

My focus on coordination in this paper is therefore induced by the dearth of research and insights on this organising problem that directly confronts GSD. Organisation research literature is replete with many constructs of coordination. However, none of these coordination constructs pay particular attention to the general problems of distributed organising or to the peculiar problems of GSD. Besides, none of these constructs represents a unified conceptual framework that is fit for analysing coordination in GSD.

To address these issues, this paper proposes a conceptual framework that will hopefully proffer a theoretical foundation for research on the coordination of GSD. The deliberations that will lead to this framework are inspired by the following research question: “What is the significance of distribution in Global Software Development, and how does it condition coordination in this domain?”

First, drawing upon existing literature on coordination, I conceptualise coordination as “managing interdependencies,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GSD'06, May 23, 2006, Shanghai, China.

Copyright 2006 ACM 1-59593-085-X/06/0005...\$5.00.

uncertainties and equivocalities, conflicts, technology representations, and their interrelations. Second, pointing to distribution as a significant conditioner of GSD, I propose, through the framework, that the distribution-engendered conditioners – geographical distance, socio-culture and technology – would significantly condition the management of process interdependencies, interpersonal and interunit conflicts, information uncertainties and equivocalities, and technology representations.

## 2. PROBLEMS WITH EXTANT COORDINATION CONSTRUCTS

Coordination is not a new concept in organisation research. It was the underpinning philosophy of Taylorism, and has remained the preoccupation of organisational researchers who have grappled with it over the years [e.g. 18, 20, 29, 33, 34]. While it is not my aim to detail every single attempt at the theorization of coordination in this paper, I will review the key attempts at theorizing coordination with the aim of pointing out the insufficiencies of existing perspectives in terms of the methodological problem of analyzing coordination in GSD.

The theorization attempts of coordination show signs of divergence and convergence of understandings. On the one hand, a careful browse through the coordination literature reveals several divergent perspectives on the problem. While many researchers [e.g. 18, 33, 34] see the problem in terms of the management of task interdependencies, others [e.g. 7] see it in terms of managing information uncertainties and equivocalities. Some [e.g. 30, 35] have theorized it in terms of interpersonal and interunit conflict management; while researchers in Computer-Supported Cooperative Work (CSCW) [e.g. 29] see it in terms of the design and utility of technology and technology representations. On the other hand, Theorists have focused more attention on interdependencies than uncertainties or conflicts because information uncertainties, inter-unit or interpersonal conflicts, and technology representations only manifest when the tasks of organisational units are interdependent. No interdependencies, no conflicts, no uncertainties, and no technology representations. As a result, “interdependence” has experienced continuous conceptual clarity over the years, judging from Thompson’s [33] exegesis, through Van de Ven and his colleagues’ [34] formulation, to Malone and Crowston’s [18] definition.

In sum, the processes that have simplified coordination as “managing interdependencies” have, at the same time, led to the gradual oversight of the other factors such as uncertainties and equivocalities, technology representations and conflict as integral problems of coordination. Stated differently, the selection and retention of interdependence in the theorization processes, apart from reinforcing interdependence as the key factor, has led to its isolation and has automatically relegated the other factors to secondary statuses often leading to their segregation.

The literature on coordination in software development [e.g. 14, 22] and global software development [e.g. 10, 12] are not exempt from the problems of divergence, convergence and omissions associated with general organisation theory literature on coordination. In this body of literature, while there is a general agreement that coordination of software development efforts are escalated when development is distributed, approaches to analyses reflect the divergent perspectives of coordination. For instance, Nidumolu [22] dwells on uncertainty as the key factor of the

coordination problem in software development; Kraut and Streeter [14] focus on uncertainty and interdependence in their theorization of coordination in software development; Herbsleb and Grinter [12] centre their analysis on formal and informal communications among distributed developers; and although Grinter and her colleagues’ [10] elucidation of four models for coordinating distributed software development embody processes, people and technology, none of their models integrates interdependencies, uncertainties and equivocalities, conflicts and technology representations.

The problem is that a coherent framework for analyzing coordination in GSD that integrates all these diverse perspectives – interdependence, uncertainties and equivocalities, conflict and technology representations – is lacking. It is apparent that earlier individual attempts at theorizing coordination dwelt on some of these perspectives at the neglect of the others. It is also apparent that in these attempts, researchers have failed to identify the key dimensions of organising software development as well as the characteristics of those dimensions. Even in instances where researchers have appreciated the increased complexity of coordination in the organisation of distributed software development [e.g. 10, 12], the needed integrated framework that embodies all the organising dimensions and their characteristics has been lacking.

To be able to properly analyse coordination in GSD, an understanding of the key elements of distribution and how they condition coordination to make it more complex in the domain are critical necessities.

## 3. DISTRIBUTION-ENGENDERED CONDITIONERS OF GSD

The foremost conditioners to highlight are the distance-related ones. In the context of the aims of this paper, the primary component is geographic separation of developers, of development processes, of technology, and of information. Furthermore, associated parameters such as distance between distributed locations in the development effort [1] and the mobility of developers [23] are worthy considerations. The distribution of the development activity, the distance between locations, and the mobility of developers can individually or collectively be understood along the dimensions of space, time and context. That is to say, the spatial, temporal and contextual dimensions give more meaning to distribution, distance and mobility, and to the coordination processes and mechanisms that are adopted to effectively organise GSD.

Next are the socio-cultural conditioners. In a GSD activity, there is no doubt that peculiar socio-cultural issues identify the developers within particularly locations in the distribution. The degrees of socio-cultural influences can be very minimal if the developers and locations in the distribution are all characterised by common socio-cultural characteristics. On the contrary, they can be very pronounced and determining if they do not share such similar socio-cultural characteristics. For example, in their discussions of GSD across borders, Sundeep et. al. [28] espouse the significance of pronounced cultural differences in GSD. To them, developers’ cultural orientations are reflected in their beliefs, perceptions, and attitudes. Thus, in the organisation of a GSD activity, location-based socio-cultural conditioners such as the role of power and knowledge in the production and reproduction of cultural norms; how belief systems translate into context-bound meanings of

information and nature of knowledge; how developers perceive reward systems and their process or outcome targets; the interpretation of modes of behaviour and outcome control; and the forms of organising in terms of markets, bureaucracies or clans [24] become important considerations.

Technology conditioners are the information and communication technologies (ICTs) that are applied in GSD and leveraged into computational coordination mechanisms [29]. In GSD, the pervading problems of distance and distribution, as well as the potential socio-cultural influences, are usually tackled with ICTs to facilitate and optimise interactions, cooperation and collaboration among distributed developers. Therefore, ICTs can be deemed as technological conditioners of a GSD that engage in a mutual interaction and shaping with the distance-related and socio-cultural conditioners. To understand technological conditioners and their impact on distributed organising [23], it is important to appreciate two broad areas of computational coordination mechanisms [29]. On the one hand, there are models of structures and processes concerning aspects such as data flows, conceptual schemes, knowledge management repositories, knowledge representations, and inscribed rules and methods [11]. On the other, there are models of presentation and access concerning issues such as user interface, functionality, ease of use and usability [2].

These distribution-engendered conditioners will bear directly on the coordination of GSD. They will sufficiently condition uncertainties and equivocalities, interdependencies, conflicts, technology representations, and their interrelationships. That is, they will condition coordination by increasing its complexity. This complexity translates into the need for a comprehensive analysis of these interrelations through thorough empirical and theoretical analysis of the coordination challenge in GSD. And this paper proposes a conceptual framework that will, hopefully, support the satisfaction of this need.

## 4. THE FRAMEWORK

### 4.1 Dimensions of GSD

One cannot analyse coordination in any domain without knowing the phenomena that are being coordinated in the first place. To begin with, it is important to appreciate the core dimensions of organising, and hence of distributed organising. To do this, I draw upon Leavitt's [15] framework of organisational change that highlights tasks, people, technology and structure as the core interacting variables of organisations. I adapt the framework to Leavitt's by highlighting people, processes (tasks), technology and information as the core dimensions of distributed organising. In the context of GSD, Leavitt's "structure" is represented by the distribution of software development organising efforts. And in recognition of the knowledge-intensive nature of software development, "information" is deemed as an equally significant dimension on the same level as people, processes and technology. The significance of information also lies in the crucial nature of its interaction with the other three dimensions and their characteristics in software development.

#### 4.1.1 People

The "people" in a GSD activity include all those individuals who have a stake in the software development activity. However, understanding the coordination problem makes the distributed

developers – programmers, testers, etc – whose concerted efforts require coordination more relevant. They are the people who matter, so to speak, when managers aim to achieve effective coordination of software development processes; and hence the processes they engage in, the technologies they adopt and use in these processes, as well as the range of information they generate, process and transmit with these technologies must be analysed. In a distributed development environment, people may be affected by the socio-cultural characteristics of the particular locations in which they operate. These may condition their belief systems, reflect in their "frames of reference" [9], reflect in their attitudes, and subsequently engender conflicts with corollaries on interdependent relationships.

#### 4.1.2 Processes

Howard Baetjer Jr's [3] observation underlines the criticality of the software development process:

"The process is a dialogue in which the knowledge that must become the software is brought together and embodied in the software. The process involves interaction between users and designers, between users and evolving tools, and between designers and evolving tools." (p.85).

Processes include all the tasks undertaken by the people such as modelling, programming and testing; all modes of interactions between them, including human-technology interactions; and information generation, processing and transmission tasks [see 13]. Software processes which are captured by phrases such as "mode of operation" [19] also have to be taken into account and judged against other essential parameters such as task complexity, uncertainties and equivocalities of requirements information, and the envisaged means of expression of the final product. Thus, analysis must centre on how distribution shapes development processes such as the life-cycle or linear-sequential modelling [25], prototyping, incremental modelling and/or spiral modelling [4].

#### 4.1.3 Information

The resource essence of information is undoubted, and its relevance as a core dimension of GSD lies in its role as the lifeblood that circulates to ensure the functional relationship between people, processes and technology. It is the source of the knowledge which Baetjer Jr. used to substantiate his arguments for the criticality of the software development process. It is the resource upon which characteristics such as uncertainties and equivocalities, and technology representations gain their meaning and substance. By information, analysis must aim to outlay aspects such as information representations, information strategies being applied, knowledge repositories and sharing modes, modes of information capture, processing and transmission, what information is being transmitted across development sites, different interpretations of information by distributed developers, and how information shapes developers' decision making. These have to be grounded in well-known elements of information such as accuracy, timeliness, reliability, completeness, sufficiency and degree of detail. These characteristics are typically embodied in information representations such as text, voice and pictures, and in this regard, theories of interpretation such as semiotics, phenomenology and hermeneutics may be applied.

#### 4.1.4 Technology

The technology dimension captures all forms of technology artefacts that are deployed in support of the interactions between people, processes and information – for example, programming languages, development platforms, bug tracking systems, and knowledge repositories. Analysis must focus on technologies such as development tools, languages and platforms, bug tracking systems and knowledge repositories on the one hand; and information and communication technologies such as various information generation, processing, and interaction systems on the other hand. Obviously, there are several instances under these main categories which must be identified in analysis. The identification must be followed up with their role in conditioning the interrelations between the users (people), processes and information. Against this background, it is important for the researcher to appreciate the structural and functional roles of any particular technology; that is, its existential and essential roles and the factors that also condition these roles. In terms of coordination, however, the functional or essential role is more challenging and interesting because in this role, technology becomes an actor..

Analysis must also distinguish between essential technologies (technologies which are intrinsic in software development) and accidental technologies (technologies which are adopted and deployed to facilitate the coordination of GSD) in order to properly appreciate the distribution-engendered factors that necessitate the deployment of those accidental technologies.

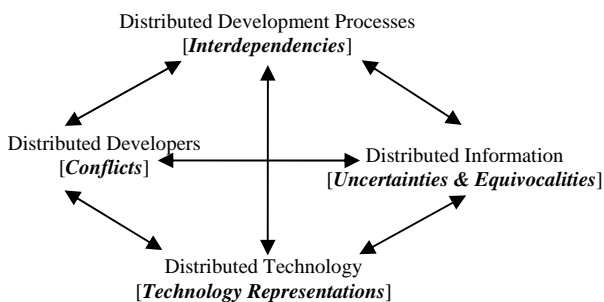


Figure 1: Conceptual Model of Coordination in GSD

## 4.2 Characteristics of Dimensions

The appreciation of these four core dimensions – people, processes, information and technology – in a GSD activity must be the starting point in analysis. It is argued that the success of GSD lies in effective coordination of these dimensions and the interactions between them. However, the mere appreciation of these dimensions will not provide the necessary insights for analysing coordination in GSD because it does not capture the *how* of the coordination challenge. These dimensions only indicate *what* will be coordinated in a GSD. Since the question of ‘how these dimensions and their interactions will be coordinated’ must be central in analysis, it is imperative to proceed beyond them to appreciate their characteristics. The ‘how’ of the coordination challenge can be appreciated only by understanding, first, how these characteristics and their interrelations are managed; and second, how their management is conditioned by distance-engendered conditioners.

In the framework, it is argued that the core characteristics of people are *conflicts* [32, 30]; those of processes are *interdependencies* [18, 17, 33]; those of information are *uncertainties* and *equivocalities* [21, 7]; and, those of technology are *technology representations* [39, 27]. Note that just as each organising dimension must be understood in terms of its interaction with the three other dimensions, every characteristic must also be analysed in terms of its cyclical relationships with other characteristics.

### 4.2.1 Process Interdependencies

Interdependence is one longstanding premise upon which coordination has been theorized in the literature. The coordination-related propositions of Thompson [33], Van De Ven *et. al.* [34], McCann and Ferry [20], Victor and Blackburn [35] and Thibaut and Kelley [31] all underline interdependence as central in coordination. Although each of these authors approached interdependence from a different perspective, its fundamental representation of a mutual relationship between two entities in an organising effort still holds true.

Interdependence is a continuous variable, and therefore an understanding of its degrees and variations in degrees over the course of a distributed activity must be taken into account in analysis. It is also necessary to understand how the other characteristics – uncertainties and equivocalities, conflicts and technology representations – would shape interdependencies. How these interdependent relationships between distributed tasks are managed, as a means of coordinating the organising dimensions and their interactions, is the essence of the framework. For instance, all the interrogative pronouns – what (management actions), how (methods), where (locations of actions), when (times of actions), and under what circumstance (contexts) – may be applied to understand how interdependencies are managed.

Most importantly, the impact of distance-related, socio-cultural influences and technology conditioners on the management of existing interdependencies must be analysed. To this end, each of management actions, methods, locations of actions, and their times can be understood in terms of each of distance-related, socio-cultural and technology conditioners.

### 4.2.2 Interpersonal and Interunit Conflicts

The direct sources of conflicts can be found in contradictory motives [16], contradictory frames of reference [9], and goal incompatibility [30] between two people or groups that relate interdependently in an organising effort. The distinction between contradictions in the motivational (mental) sphere and actual conflict at the physical level (“blocking” [30]) is important because it automatically necessitates the analyst’s attentiveness to those issues that motivate people to engage in conflicts. It is also important in the sense that an awareness of those motivating issues can facilitate an understanding of the conflict management measures being instituted or implemented in a GSD activity.

The consequences of conflicts in a GSD activity are as important as the antecedents because, like the management of uncertainties and equivocalities, there is the likelihood of circular causation between the consequences and antecedents of conflicts. This may reinforce existing conflicts, transmute them, or generate entirely new ones if their management is not directed at both the consequences and antecedents. In GSD, some people’s motives may be directly linked with the socio-cultural norms of the

particular locations in which they work, or where they have been nurtured, for example. Walsham's [37] analysis of cultural-based structural contradictions and conflicts between globally-distributed software teams clearly attests to this reality. Conflicts between any parties may also be understood from the organisational perspective. That is, in analysis, it is important to understand the nature of organising (centralized and formalized) and control (frequency of surveillance and severity of enforcement), and their impact on employee satisfaction and conflict [8, 38]. Conflict may also be linked with one party's perception of an opportunity for interference by another party. Perceived opportunity for interference is a potential occurrence in scenarios where the two parties share a common resource or each one's outputs constitute inputs for the other (mutual interdependence).

To this end, the analyst must aim to understand how conflict is managed in a GSD activity. He or she must pay attention to the areas where the management efforts are directed – at the antecedents or consequences or both. It is also important to understand the interrelationships between conflicts and the other characteristics – uncertainties and equivocalities, interdependence and technology representations. But most importantly, analysis must aim to understand how each of distance-related, socio-cultural and technological conditioners shape conflicts, their antecedents and consequences, their interrelationships with the three characteristics, and their management.

#### 4.2.3 Information Uncertainties and Equivocalities

Information uncertainty is an entity's inability to predict information about a phenomenon accurately [7, 21 p.136]. Information equivocality, on the other hand, relates to the ambiguity of some information and its sources faced by the entity. First and foremost, it is important to distinguish between what is meant by uncertainties here and Mathiassen and Stage's [19] notion of uncertainties which they deem as an essential attribute of software development. Uncertainty as an essence of software development is related to requirements information, and must therefore be understood in terms of the problems of requirements elicitation and analysis instead of the organisational understanding espoused, for example, by Milliken [21] and Daft and Lengel [7]. The latter understanding – the one with organisational overtones – is central in the conceptualization of the framework. In line with the same disposition, equivocality signifies the difficulty in deciphering the right information from ambiguous information from individuals, groups or departments in an organisation [7]; its meaning here does not principally concur with that of Mathiassen and Stage [19].

I am arguing for uncertainties and equivocality to be integrated with interdependencies, conflicts and technology representations because information is the lifeblood of the interrelationships between people, processes and technology. In software development, interactions between people, for example, is considered as one of the dominant processes. These interactions may be conducted face-to-face, thus enabling the flow of "rich information" between interacting parties. In GSD, however, technology-mediated interactions would, most likely, be used as proxies for face-to-face interactions. Drawing upon Daft and Lengel's [6] information richness theory, information exchanged through technology-mediated interactions are generally "poorer" than those exchanged in face-to-face interactions. Since technology-mediated interactions are virtually inevitable in the coordination of GSD, poor information also seems inevitable in

the domain. Uncertainties and equivocality are direct instances of the poverty of information that come with technology-mediated interactions, and their interrelationships with the other characteristics in GSD must be well-appreciated by the analyst.

Appreciating these interrelationships is a necessary basis for probing for answers to the degree to which distance-related, socio-cultural and technological conditioners shape information uncertainties and equivocality.

Probing should not end there: in a GSD, the analyst must also explore how uncertainties and equivocality are being managed. Typically, uncertainty problems are managed by generating more information to make oneself more certain of the state, cause or effect of a phenomenon. Conversely, problems of equivocality are managed through information processing as means of making available information more unambiguous. The interesting feature of the relationship between uncertainties and equivocality is that there is the likelihood for their individual managements to relate in circular causation. That is to say, information generation as a measure for managing uncertainties may lead to new equivocality, which management through information processing may cause entropy and new uncertainties, which may require further information generation, and so on. Therefore, it is important that the analyst concerns him- or herself with how each of distance-related, socio-cultural and technological conditioners shapes these uncertainties and equivocality management efforts plus the circular causation between them.

#### 4.2.4 Technology Representations

By technology representations, I mean the role of technology in the software development effort. For example, for every unit of technology deployed in the development effort, the analyst must beware of the instances in which it acts as a product and those in which it acts as a service. That is; what are the circumstances in which it plays a structural role and what are those in which it plays a functional role? "Representations are social facts" [26], and technology representations connote the socio-technical role either assigned to or obtainable from any piece of technological artefact in an activity.

To understand the socio-technical implications of technology, it is important to understand technology (hardware and software) and their representations (structural and functional) as tools and signs that, according to Vygotsky [36], are mutually interrelated and separate at the same time:

"[The] most essential difference between sign and tool, and the basic real divergence of the two lines, is the different ways that they orient human behaviour. The tool's function is to serve as the conductor of human influence on the object of activity; it is *externally* oriented; it must lead to changes in objects. ... The sign, on the other hand, changes nothing in the object of a psychological operation. It is a means of internal activity aimed at mastering oneself; the sign is *internally* oriented." (p.55) (italics in original).

Thus, on the one hand, the tool representation must be understood in terms of how it facilitates people's efforts in a software development activity (external orientation of technology representations); and, on the other hand, the sign representation must be understood in terms of how it shapes people's psychological attributes such as attitudes, feelings, perceptions, motives and frames of reference. The latter is important for understanding the antecedents of conflict, the potential roles of

**Table 1: Framework for Analyzing Coordination in GSD.**

		Distribution-engendered Conditioners		
		Distance-related	Socio-Cultural	Technological
		<ul style="list-style-type: none"> <li>Distance (spatial, temporal, contextual, etc)</li> <li>Mobility (spatial, temporal, contextual, etc)</li> <li>Etc</li> </ul>	<ul style="list-style-type: none"> <li>Cultural norms (power, knowledge)</li> <li>Belief systems</li> <li>Reward systems</li> <li>Context-bound meanings of information</li> <li>Etc</li> </ul>	<ul style="list-style-type: none"> <li>Remote interaction technologies</li> <li>Inscriptions</li> <li>Computer-based coordination mechanisms</li> <li>Etc</li> </ul>
Management of:	<b>Interdependencies</b> <ul style="list-style-type: none"> <li>Actions (what)</li> <li>Methods (how)</li> <li>Locations (where)</li> <li>Times (when)</li> <li>Contexts(in what circumstances)</li> <li>Etc</li> </ul>	<p><b>General Research Question:</b></p> <p><i>In what ways do distance, socio-culture and technology condition the management of process interdependencies, interpersonal and interunit conflicts, information uncertainties and equivocalities, and technology representations?</i></p>		
	<b>Conflicts</b> <ul style="list-style-type: none"> <li>Antecedents</li> <li>Consequences</li> <li>Circular causality</li> <li>Etc.</li> </ul>			
	<b>Uncertainties and Equivocalities</b> <ul style="list-style-type: none"> <li>Information generation</li> <li>Information processing</li> <li>Circular causality</li> <li>Etc.</li> </ul>			
	<b>Technology Representations</b> <ul style="list-style-type: none"> <li>Functional assignment</li> <li>Rules inscriptions</li> <li>Technology as actant</li> <li>Etc.</li> </ul>			

equivocal and uncertain information, and the management of technology representations.

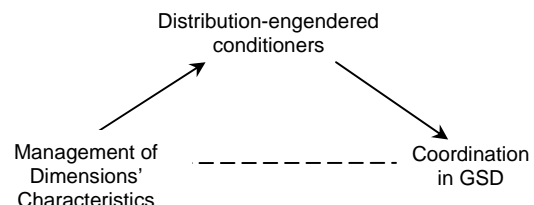
The knowledge that software development is itself dependent on computer-based hardware and software (the essential technologies) is basic; however, additional technologies are introduced to coordinate the interrelations between people, development processes and information (the accidental technologies). In the conceptualization of the characteristics of the technology dimension, the essential technologies and their representations would be more interesting. That is to say, inscribed behavioural standards in information infrastructure [11] that transform technologies into representations would be interesting.

However, as I have reiterated in this paper, understanding how technology representations mutually interrelate with interdependencies, conflicts and uncertainties and equivocalities, and how they are managed are more important than merely understanding technology representations. It is fundamentally wrong to isolate technology representations because the representations can assume the roles of any of the organising dimensions – people, processes and information – in a software development activity. Against this background, analysis must aim to understand the particular circumstances surrounding software development that instigate any technology unit to assume any particular representation.

While the mutual interrelationships with the other characteristics are important for understanding technology representations in software development, the role of the conditioners – distance-related, socio-cultural and technological – in shaping these representations in a GSD is also important for understanding coordination in GSD. And in this regard, the analyst must eventually orient towards the latter understanding.

## 5. SUMMARY

The purpose of this paper was to integrate a number of segregated coordination constructs, to relate them with distributed-engendered conditioners of GSD, and to emerge with a framework for the analysis of coordination in GSD (see Table 1). The framework conveys the following understanding: in GSD, the management of process interdependencies, interpersonal and interunit conflicts, information uncertainties and equivocalities, and technology representations will be significantly conditioned by distance, socio-culture and technology (see Figure 2). To get to the bottom of this understanding, the analyst must aim at addressing the general research question posed in Table 1 with the ultimate aim of understanding *how* coordination of GSD activities would be achieved in the face of distance-related, socio-cultural and technological challenges.



**Figure 2: Conceptualization of Coordination in GSD**

One fundamental attribute of every conceptualization effort and outcome is its abstraction, which is a weakness as far as empiricism is concerned. My arguments in this paper are grounded on theoretical information drawn from the literature; that is to say, they lack the basis of a full empirical example backed by a sound methodology. However, the framework is not meant to be a

theoretical solution; on the contrary, it is meant to draw researchers' attention to some issues which have been overlooked in the literature, and which are necessary for analyzing coordination in GSD. Furthermore, the framework is not exempt from the potential problems of frameworks – their likelihood to enslave their users. Therefore, I would not advice that it be applied uncritically. These limitations notwithstanding, I believe this framework and its underpinning arguments have provided considerable answers to the initial question – “What is the significance of distribution in Global Software Development, and how does it shape coordination in this domain?” These contributions are the first steps towards progress in research on the coordination of GSD; and they are subject to scrutiny through further research in the area by all concerned.

## 6. ACKNOWLEDGEMENTS

This paper is a product from research that is funded by Science Foundation Ireland.

## 7. REFERENCES

- [1] Abbott, P. and Jones, M.R. The Importance of being Nearest: Nearshore Software Outsourcing and Globalisation Discourse. In *Proceedings of the IFIP TC8/WG 8.2*. Barcelona. 2002.
- [2] Adams, D.A., Nelson, R.R., and Todd, P.A. Perceived Usefulness, Ease of Use, and Usage of Information Technology: A Replication. *MIS Quarterly*, 16, 2: 227-247. 1992.
- [3] Baetjer Jr, H. *Software as Capital: An Economic Perspective on Software Engineering*. IEEE Computer Society Press. 1998.
- [4] Boehm, B. A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21, 5: 61-72. 1988.
- [5] Carmel, E. and Tjia, P. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge University Press, Cambridge. 2005.
- [6] Daft, R.L. and Lengel, R.H. *Information Richness: A New Approach to Manager Information Processing and Organizational Design*, in: *Research in Organizational Behaviour*, B. Straw and L.L. Cummings (eds.) JAI Press: Greenwich, CT. 1984
- [7] Daft, R.L. and Lengel, R.H. Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, 32, 5: 554-571. 1986.
- [8] Dewar, R. and Werbel, J. Universalistic and Contingency Predictions of Employee Satisfaction and Conflict. *Administrative Science Quarterly*, 24: 426-448. 1979.
- [9] Gioia, D.A. *Symbols, Scripts, and Sensemaking: Creating Meaning in the Organizational Experience*, in: *The Thinking Organization*, H.P. Sims and D.A. Gioia (eds.) Jossey-Bass: San Francisco, CA. 1986
- [10] Grinter, R.E., Herbsleb, J.D., and Perry, D.E. The Geography of Coordination: Dealing with Distance in R&D Work. In *Proceedings of the GROUP'99*. Phoenix, Arizona. 1999.
- [11] Hanseth, O. and Monteiro, E. Inscripting Behaviour in Information Infrastructure Standards. *Accounting, Management and Information Technologies*, 7, 4: 183-211. 1997.
- [12] Herbsleb, J.D. and Grinter, R.E. Splitting the Organization and Integrating the Code: Conway's Law Revisited. In *Proceedings of the ICSE'99*. Los Angeles, CA. 1999.
- [13] Humphrey, W.S. *Managing the Software Process*. Addison-Wesley Professional, Reading, MA. 1989.
- [14] Kraut, R.F. and Streeter, L.A. Coordination in Software Development. *Communications of the ACM*, 38, 3: 69-81. 1995.
- [15] Leavitt, H.J. *Applied Organisational Change in Industry: Structural, Technological and Humanistic Approaches*, in: *Handbook of Organizations*, J.G. March (ed.) Rand McNally: Chicago. 1965
- [16] Leont'ev, A.N. *Problems of the Development of the Mind*. Progress Publishers, Moscow. 1982.
- [17] Malone, T.W. and Crowston, K. *What is Coordination Theory and How Can It Help Design Cooperative Work Systems?*, in: *Proceedings of the 3rd Conference on Computer-Supported Cooperative Work* ACM Press: New York. 1990
- [18] Malone, T.W. and Crowston, K. The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26, 1: 87-119. 1994.
- [19] Mathiassen, L. and Stage, J. The Principle of Limited Reduction in Software Design. *Information Technology & People*, 6, 2-3: 171-185. 1992.
- [20] McCann, J.E. and Ferry, D.L. An approach for Assessing and Managing Inter-unit Interdependence. *Academy of Management Review*, 4, 1: 113-119. 1979.
- [21] Milliken, F.J. Three Types of Perceived Uncertainty about the Environment: State, Effect, and Response Uncertainty. *Academy of Management Review*, 12: 133-143. 1987.
- [22] Nidumolu, S.R. The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable. *Information Systems Research*, 6, 3: 191-219. 2001.
- [23] Orlikowski, W.J. Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. *Organization Science*, 13, 3: 249-273. 2002.
- [24] Ouchi, W.G. Markets, Bureaucracies and Clans. *Administrative Science Quarterly*, 25: 129-141. 1980.
- [25] Pressman, R.S. *Software Engineering: A Practitioner's Approach*. McGraw Hill, Berkshire, UK. 2000.
- [26] Rabinow, P. *Representations are Social Facts: Modernity and Post-Modernity in Anthropology*, in: *Essays in the Anthropology of Reason*, P. Rabinow (ed.) Princeton University Press: Princeton, NJ. 1998
- [27] Robinson, M. and Bannon, L. Questioning Representations. In *Proceedings of the European Computer-Supported Cooperative Work Conference (ECSCW'91)*. Amsterdam. 1991.
- [28] Sahay, S., Nicholson, B., and Krishna, S. *Global IT Outsourcing: Software Development Across Borders*. Cambridge University Press, Cambridge, UK. 2003.
- [29] Schmidt, K. and Simone, C. Coordination Mechanisms: Towards a Conceptual Foundation of Computer Supported Cooperative Work Systems Design. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 5: 155-200. 1996.
- [30] Schmidt, S.M. and Kochan, T.A. Conflict: Toward Conceptual Clarity. *Administrative Science Quarterly*, 17, 3: 359-370. 1972.
- [31] Thibaut, W.J. and Kelley, H.H. *The Social Psychology of Groups*. Wiley, New York. 1959.
- [32] Thompson, J.D. Organizational Management of Conflict. *Administrative Science Quarterly*, 4: 389-409. 1960.
- [33] Thompson, J.D. *Organizations in Action*. McGraw Hill, Chicago. 1967.
- [34] Van de Ven, A.H., Delbecq, A.L., and Koenig, R. Determinants of Coordination Modes within Organizations. *American Sociological Review*, 41, 2: 322-328. 1976.
- [35] Victor, B. and Blackburn, R.S. Interdependence: An Alternative Conceptualisation. *Academy of Management Review*, 12, 3: 486-498. 1987.
- [36] Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA. 1978.
- [37] Walsham, G. *Globalization and ICTs: Working Across Cultures*. Working Paper #8, Judge Institute of Management, University of Cambridge, Cambridge, UK.
- [38] Walton, R.E. and Dutton, J.M. The Management of Interdepartmental Conflict: A Model and Review. *Administrative Science Quarterly*, 14: 73-84. 1969.
- [39] Wartofsky, M.W. *Models: Representation and Scientific Understanding*. Reidel, Dordrecht. 1979.
- [40] Wiredu, G.O. Coordination as the Challenge of Distributed Software Development. In *Proceedings of the International Workshop on Distributed Software Development (DiSD'05)*. Paris, France. 2005c.